Robust Tensor Completion via Capped Frobenius Norm

Xiao Peng Li, Zhi-Yong Wang, Zhang-Lei Shi, Hing Cheung So, *Fellow, IEEE*, and Nicholas D. Sidiropoulos, *Fellow, IEEE*

***Abstract*— Tensor completion (TC) refers to restoring the missing entries in a given tensor by making use of the low-rank structure. Most existing algorithms have excellent performance in Gaussian noise *or* impulsive noise scenarios. Generally speaking, the Frobenius-norm-based methods achieve excellent performance in additive Gaussian noise, while their recovery severely degrades in impulsive noise. Although the algorithms using the ℓ_p-norm ($0 < p < 2$) or its variants can attain high restoration accuracy in the presence of gross errors, they are inferior to the Frobenius-norm-based methods when the noise is Gaussian-distributed. Therefore, an approach that is able to perform well in both Gaussian noise and impulsive noise is desired. In this work, we use a capped Frobenius norm to restrain outliers, which corresponds to a form of the truncated least-squares loss function. The upper bound of our capped Frobenius norm is automatically updated using normalized median absolute deviation during iterations. Therefore, it achieves better performance than the ℓ_p-norm with outlier-contaminated observations and attains comparable accuracy to the Frobenius norm without tuning parameter in Gaussian noise. We then adopt the half-quadratic theory to convert the nonconvex problem into a tractable multivariable problem, that is, convex optimization with respect to (w.r.t.) each individual variable. To address the resultant task, we exploit the proximal block coordinate descent (PBCD) method and then establish the convergence of the suggested algorithm. Specifically, the objective function value is guaranteed to be convergent while the variable sequence has a subsequence converging to a critical point. Experimental results based on real-world images and videos exhibit the superiority of the devised approach over several state-of-the-art algorithms in terms of recovery performance. MATLAB code is available at https://github.com/Li-X-P/Code-of-Robust-Tensor-Completion.**

***Index Terms*— Capped Frobenius norm, proximal block coordinate descent, robust recovery, tensor completion (TC), tensor ring.**

Manuscript received 9 August 2022; revised 14 December 2022; accepted 1 January 2023. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project CityU 11201421 and Project CityU 11207922; and in part by the Fundamental Research Funds for the Central Universities under Grant 22CX06039A. *(Corresponding authors: Zhi-Yong Wang; Hing Cheung So.)*

Xiao Peng Li, Zhi-Yong Wang, and Hing Cheung So are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, SAR, China (e-mail: x.p.li@my.cityu.edu.hk; z.y.wang@my.cityu.edu.hk; hcso@ee.cityu.edu.hk).

Zhang-Lei Shi is with the College of Science, China University of Petroleum (East China), Qingdao 266580, China (e-mail: zlshi@upc.edu.cn).

Nicholas D. Sidiropoulos is with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904 USA (e-mail: nikos@virginia.edu).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TNNLS.2023.3236415.

Digital Object Identifier 10.1109/TNNLS.2023.3236415

I. Introduction

TENSOR (a.k.a. multiway array) is the multidimensional extension of scalar, vector, and matrix [1] and can represent many real-world signals, including color images, videos, hyperspectral images, and radar data to name a few. Although tensors can be unfolded into matrices and then be processed by matrix techniques [2], [3], this procedure may discard the inherent information in the high order. Therefore, multilinear algebra for tensors is a powerful tool to analyze high-order data.

The success of low-rank matrix recovery [4], [5] has inspired a large number of researchers to expand the concept to tensor completion (TC) for processing high-order data. TC aims at recovering the missing entries of a partially observed tensor using low-rank structure and has a wide range of applications, such as image inpainting [6], video restoration [7], multiple-input multiple-output (MIMO) radar localization [8], object detection [9], background–foreground separation [10], [11], and hyperspectral image recovery [12]. This is because the dominant information of high-dimensional data is contained in its low-dimensional subspace.

Analogous to matrix completion, TC is intuitively formulated as a rank minimization problem with the constraint that the recovered and given tensors are identical in the observation set. Since it is difficult to handle the rank minimization problem, most existing algorithms adopt the low-rank factorization strategy to achieve TC. Note that various tensor decomposition models generate different TC methods. For instance, CANDECOMP/PARAFAC (CP) decomposition factorizes a tensor into a sum of outer products of vectors [13], [14], which has been applied to tensor restoration [15], [16], [17]. The Tucker decomposition uses one small core tensor and a set of matrices [18], and the corresponding TC algorithms include [19], [20]. Besides, tensor singular value decomposition (t-SVD) [21] has been adopted for tensor recovery [22], [23], [24]. In addition, tensor train factorization decomposes an Nth-order tensor into two matrices and $(N-2)$ third-order tensors [25]. It has also been exploited for TC, and it demonstrates better recovery performance than the CP decomposition, Tucker decomposition, and t-SVD [7], [26], [27]. As an improvement of the tensor train factorization, tensor ring decomposition factorizes an Nth-order tensor into N third-order tensors [28] and has been applied to tensor recovery [29], [30], [31]. On the other hand, a fully connected tensor network decomposition has been developed to restore higher order tensors [32]. Compared with tensor train and tensor ring factorizations, it decomposes an Nth-order tensor into N Nth-order tensors, which results in a more adequate and flexible representation.

Despite the fact that Gaussian distribution is the most typical noise model, non-Gaussian-distributed noise also appears in real life [33], for example, salt-and-pepper noise in visual data [34] and impulsive noise in communication channels [35]. The aforementioned methods adopt the Frobenius norm as their loss function and thus they are not robust to gross errors. Therefore, their performance might be degraded when the observed tensor is corrupted by the outliers. To restrain anomalies, the ℓ_p-norm with $p \in (0, 2)$ has been applied to minimize the recovered error, resulting in iteratively reweighted t-SVD (IR-t-SVD) [36], ℓ_p-regression tensor train completion (ℓ_p-TTC) [37], and trilinear alternating least absolute error regression (TALAE) [38]. Nevertheless, the ℓ_p-norm-based methods have relatively high computational complexity since solving the ℓ_p-norm-based problem requires a multilayer iterative procedure. Besides, the ℓ_p-norm with $p \in (0, 1)$ poses a challenge because it is a nonconvex and nonsmooth function. To tackle this issue, Chen et al. [39] propose a logarithmic norm to approximate the ℓ_p-norm with $p \in (0, 1)$ and apply it for tensor recovery in the presence of gross errors, yielding an algorithm called logarithmic norm minimization and outlier projection (LNOP). Li and So [40] suggest approximating the ℓ_p-norm with the $\ell_{p,\epsilon}$-norm, which demonstrates better performance than the ℓ_p-norm in robust TC (RTC). Moreover, the robust principal component analysis (RPCA) concept [41] has been used for RTC, resulting in the collaborative sparse and low-rank transforms model (CSLRT) [42] and robust tensor ring completion (RTRC) [43].

Although the ℓ_p-norm and its variants exhibit superior performance over the Frobenius norm in the presence of the outliers, the former is inferior to the latter in the Gaussian noise scenarios. Moreover, the ℓ_p-norm applies to whole tensor, implying that all the elements are considered as anomaly-contaminated. However, only a small proportions are typically corrupted by anomalies, and thus its performance is not optimal in impulsive noise. Conceptually, the RPCA model attains excellent performance in both the white Gaussian noise and gross error scenarios. Nevertheless, in practice, its solution depends on the choice of the regularization parameter. That is, for different data and noise, practitioners need to carefully adjust the penalty parameter to achieve good performance, which is time-consuming.

In this work, our aim is to devise an RTC method to suppress the white Gaussian noise and/or impulsive noise without the need to tune any parameter. We address RTC using the idea of the capped Frobenius norm, corresponding to a form of the truncated least-squares loss function [44], [45]. The interpretation of the upper bound of this norm is the boundary between the normal and outlier-contaminated entries. Therefore, we exploit the normalized median absolute deviation strategy to update the bound, such that the capped Frobenius norm is capable of effectively resisting the white Gaussian noise and/or impulsive noise. We then combine the capped Frobenius norm with tensor ring decomposition to tackle RTC. Although the capped Frobenius norm is nonconvex, we use the half-quadratic theory [46], [47] to convert the resultant problem into a tractable form, that is, convex optimization with respect to (w.r.t.) each individual variable. Subsequently, we adopt the proximal block coordinate descent (PBCD) method [48] to handle the multivariable optimization in which one block is updated while the remaining blocks are fixed at each iteration. Moreover, we establish the convergence of the proposed algorithm. Specifically, the suggested method ensures the objective function value to converge and the variable sequence to have a subsequence converging to a critical point. Our main contributions are summarized as follows.

1) We adopt the normalized median absolute deviation to obtain a robust standard deviation of the fitting error. Then, given a confidence interval, the upper bound is adaptively determined. Therefore, the capped Frobenius norm achieves excellent performance in the presence of Gaussian noise and/or impulsive noise.
2) We simplify the capped Frobenius norm based formulation into a Frobenius norm optimization with a regularization term. Then, PBCD is adopted as the solver, and all the subproblems have closed-form solutions.
3) We analyze the convergence behavior of the proposed algorithm. We prove that the objective function value is guaranteed to be convergent while the variable sequence contains a subsequence converging to a critical point.
4) Experimental results using real-world images and videos demonstrate that the devised approach is superior to popular robust methods in the presence of impulsive noise. In addition, without tuning any parameter, the performance of our method approaches that of the Frobenius-norm-based approach in the white Gaussian noise scenarios.

The remainder of this article is organized as follows. In Section II, we introduce notations and preliminaries. The proposed algorithm is presented in Section III. Besides, we analyze its convergence behavior and computational complexity. In Section IV, numerical examples are included to evaluate the devised method by comparing with several state-of-the-art algorithms. Finally, concluding remarks are given in Section V.

II. Background

In this section, notations and basic definitions are provided, and relevant works are reviewed.

A. Notations

Scalars, vectors, matrices, and tensors are denoted by italic, bold lower case, bold upper case, and bold calligraphic letters, respectively. For example, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ signifies an Nth-order tensor, and its $(l_1, l_2, \ldots, l_N)$ entry is denoted by $x_{l_1,l_2,\ldots,l_N}$ or $\boldsymbol{\mathcal{X}}(l_1, l_2, \ldots, l_N)$. The mode-$n$ unfolding of $\boldsymbol{\mathcal{X}}$ is represented by $\boldsymbol{\mathcal{X}}_{[n]} \in \mathbb{R}^{I_n \times (I_1 \cdots I_{n_1} I_{n+1} \cdots I_N)}$. Given a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, its Frobenius norm and ℓ_p-norm with $p \in (0, 2)$ are $\|\boldsymbol{\mathcal{X}}\|_F = (\sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} x_{i,j,k}^2)^{1/2}$ and $\|\boldsymbol{\mathcal{X}}\|_p = (\sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} x_{i,j,k}^p)^{1/p}$, respectively. Consider a matrix $\boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2}$, $\boldsymbol{X}^T$ is the transpose of $\boldsymbol{X}$, and $\mathrm{tr}(\boldsymbol{X}) = \sum_{i=1}^{\min(I_1,I_2)} x_{i,i}$ is the trace of $\boldsymbol{X}$. The vectorization operator

TABLE I
LIST OF NOTATIONS

Symbol	Description
x	scalar
$\boldsymbol{x} \in \mathbb{R}^{I}$	vector
$\boldsymbol{X} \in \mathbb{R}^{I_1 \times I_2}$	$I_1 \times I_2$ matrix
$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$	Nth-order tensor
$x_{l_1, l_2, \cdots, l_N}$ or $\boldsymbol{\mathcal{X}}(l_1, l_2, \cdots, l_N)$	element of $\boldsymbol{\mathcal{X}}$
$\boldsymbol{\mathcal{X}}(:, j, :)$	jth lateral slice of $\boldsymbol{\mathcal{X}}$
$\boldsymbol{\mathcal{X}}(i, :, k)$	vertical fiber of $\boldsymbol{\mathcal{X}}$
$\boldsymbol{\mathcal{X}}_{[n]} \in \mathbb{R}^{I_n \times (I_1 \cdots I_{n_1} I_{n+1} \cdots I_N)}$	mode-n unfolding of $\boldsymbol{\mathcal{X}}$
$\\|\boldsymbol{x}\\|_2 = \sqrt{\sum_{i=1}^{I} x_i^2}$	ℓ_2-norm
$\\|\boldsymbol{\mathcal{X}}\\|_F = \sqrt{\sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} x_{i,j,k}^2}$	Frobenius norm
$\\|\boldsymbol{\mathcal{X}}\\|_p = (\sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} x_{i,j,k}^p)^{1/p}$	ℓ_p-norm with $p \in (0, 2)$
$\\|\boldsymbol{\mathcal{X}}\\|_{CF} = \sqrt{\sum_{i=1}^{I_1} \sum_{j=1}^{I_2} \sum_{k=1}^{I_3} \min(x_{i,j,k}^2, \theta^2)}$	capped Frobenius norm
$\mathrm{tr}(\boldsymbol{X}) = \sum_{i=1}^{\min(I_1, I_2)} x_{i,i}$	trace of $\boldsymbol{X}$
$\boldsymbol{X}^T$	transpose of $\boldsymbol{X}$
$\boldsymbol{\mathcal{X}}^{T_n}$	tensor permutation of $\boldsymbol{\mathcal{X}}$
$\boldsymbol{x}_{:,j}$	jth column of $\boldsymbol{X}$
$\boldsymbol{x}_{i,:}$	ith row of $\boldsymbol{X}$
$\mathrm{vec}(\boldsymbol{X}) = [\boldsymbol{x}_{:,1}^T; \boldsymbol{x}_{:,2}^T; \cdots; \boldsymbol{x}_{:,I_2}^T]^T$	vectorization operator
$\mathrm{mat}(\mathrm{vec}(\boldsymbol{X})) = \boldsymbol{X}$	matrization operator
$\odot$	Hadamard product
$\circ$	outer product

is defined as $\mathrm{vec}(\boldsymbol{X}) = [\boldsymbol{x}_{:,1}^T; \boldsymbol{x}_{:,2}^T; \cdots; \boldsymbol{x}_{:,I_2}^T]^T$ where $\boldsymbol{x}_{:,j}$ for $j \in [1, I_2]$ stands for the jth column of $\boldsymbol{X}$, while $\mathrm{mat}(\cdot)$ is the matrization operator, such that $\mathrm{mat}(\mathrm{vec}(\boldsymbol{X})) = \boldsymbol{X}$. Moreover, $\odot$ and $\circ$ signify the Hadamard product and outer product operators, respectively. Other mathematical symbols are defined in their first appearance. The notations are summarized in Table I.

B. Basic Operations

Similar to the matrix product, the tensor product of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_3 \times I_4 \times I_5}$ is defined as $\boldsymbol{\mathcal{X}}\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times (I_2 I_4) \times I_5}$ [49], and it is computed as follows:

$$\boldsymbol{\mathcal{Z}}(:, l_2 \times l_4, :) = \boldsymbol{\mathcal{X}}(:, l_2, :)\boldsymbol{\mathcal{Y}}(:, l_4, :) \tag{1}$$

where $l_2 \in [1, I_2]$, $l_4 \in [1, I_4]$, and $\boldsymbol{\mathcal{X}}(:, l, :)$ stands for the lth lateral slice of $\boldsymbol{\mathcal{X}}$.

In addition, the tensor permutation operation of $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is represented by $\boldsymbol{\mathcal{X}}^{T_n}$ [49]. The relationship between $\boldsymbol{\mathcal{X}}^{T_n}$ and $\boldsymbol{\mathcal{X}}$ is

$$\boldsymbol{\mathcal{X}}^{T_n}(l_n, \ldots, l_N, l_1, \ldots, l_{n-1}) = \boldsymbol{\mathcal{X}}(l_1, l_2, \ldots, l_N). \tag{2}$$

Since the proposed algorithm is based on tensor ring decomposition, we briefly introduce it for completeness. Tensor ring decomposition factorizes an Nth-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ into N third-order tensors [28]

$$\boldsymbol{\mathcal{X}} = \sum_{r_1, \ldots, r_N = 1}^{R_1, \ldots, R_N} \boldsymbol{\mathcal{Y}}_1(r_1, :, r_2) \circ \boldsymbol{\mathcal{Y}}_2(r_2, :, r_3) \circ \cdots \circ \boldsymbol{\mathcal{Y}}_N(r_N, :, r_1) \tag{3}$$

where $\boldsymbol{\mathcal{Y}}_n \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$, $\boldsymbol{R} = [R_1, \ldots, R_N]$ is the multilinear tensor ring rank, and $\boldsymbol{\mathcal{Y}}_n(r_n, :, r_{n+1})$ is the vertical fiber. Besides, the entrywise formulation is

$$\boldsymbol{\mathcal{X}}(l_1, \cdot, l_N) = \mathrm{tr}\Big(\boldsymbol{\mathcal{Y}}_1(:, l_1, :)\boldsymbol{\mathcal{Y}}_2(:, l_2, :) \cdots \boldsymbol{\mathcal{Y}}_N(:, l_N, :)\Big). \tag{4}$$

In this work, (3) is represented by $\boldsymbol{\mathcal{X}} = h(\boldsymbol{\mathcal{Y}}_1 \cdots \boldsymbol{\mathcal{Y}}_N)$ for conciseness.

C. Related Work

Consider a partially observed tensor $\boldsymbol{\mathcal{M}}_{\boldsymbol{\Omega}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ where $\boldsymbol{\Omega} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is a binary tensor, consisting of 0 and 1. In $\boldsymbol{\Omega}$, $\Omega_{l_1, \ldots, l_N} = 1$ signifies that the corresponding $m_{l_1, \ldots, l_N}$ is observed, while $\Omega_{l_1, \ldots, l_N} = 0$ indicates a missing entry, namely, $m_{l_1, \ldots, l_N} = 0$. Therefore, $\boldsymbol{\mathcal{M}}_{\boldsymbol{\Omega}}$ has the form of

$$\boldsymbol{\mathcal{M}}_{\boldsymbol{\Omega}} = \boldsymbol{\mathcal{M}} \odot \boldsymbol{\Omega}. \tag{5}$$

As in matrix completion, TC can be formulated as a rank minimization problem

$$\min_{\boldsymbol{\mathcal{X}}} \ \mathrm{rank}(\boldsymbol{\mathcal{X}}), \quad \text{s.t. } \boldsymbol{\mathcal{X}} \odot \boldsymbol{\Omega} = \boldsymbol{\mathcal{M}}_{\Omega} \tag{6}$$

where $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. Since the rank function is nonconvex and discrete, Zhang et al. [50] propose using tensor nuclear norm instead of the rank function to solve (6), leading to

$$\min_{\boldsymbol{\mathcal{X}}} \ \|\boldsymbol{\mathcal{X}}\|_{\mathrm{TNN}}, \quad \text{s.t. } \boldsymbol{\mathcal{X}} \odot \boldsymbol{\Omega} = \boldsymbol{\mathcal{M}}_{\Omega} \tag{7}$$

where $\|\cdot\|_{\mathrm{TNN}}$ denotes the tensor nuclear norm, defined in [50]. Since the tensor nuclear norm is defined based on the third-order tensors, its application is limited. To handle higher order tensors, Wang et al. [29] suggest exploiting tensor ring decomposition to formulate the tensor recovery problem, resulting in

$$\min_{\boldsymbol{\mathcal{Y}}_{n:n=1,\ldots,N}} \|\boldsymbol{\mathcal{M}}_{\boldsymbol{\Omega}} - h(\boldsymbol{\mathcal{Y}}_1 \cdots \boldsymbol{\mathcal{Y}}_N) \odot \boldsymbol{\Omega}\|_F^2. \tag{8}$$

However, the above-mentioned methods are not robust to the outliers since they adopt the Frobenius norm for error minimization. Consider a linear regression problem based on the Frobenius norm

$$\min_{\boldsymbol{X}} \|\boldsymbol{A}\boldsymbol{X} - \boldsymbol{Y}\|_F^2 = \min_{\boldsymbol{x}_{:,j}} \sum_{i=1}^{I_1} \sum_{j=1}^{I_2} (\boldsymbol{a}_{i,:}\boldsymbol{x}_{:,j} - y_{i,j})^2. \tag{9}$$

If $y_{i,j}$ with $i \in [1, I_1]$ and $j \in [1, I_2]$ is corrupted by an outlier, the ratio between the corresponding residual $|\boldsymbol{a}_{i,:}\boldsymbol{x}_{:,j} - y_{i,j}|$ and that of a normal entry is severely enlarged by the Frobenius norm. Then, to minimize the whole fitting error, $\boldsymbol{x}_{:,j}$ will tend to reduce $(\boldsymbol{a}_{i,:}\boldsymbol{x}_{:,j} - y_{i,j})^2$. For instance, consider $\boldsymbol{A} = [1, 2; 3, 4]$, $\boldsymbol{X} = [-0.05, -0.3; 0.075, 0.275]$, and $\boldsymbol{Y} = [0.1, 0.25; 0.15, 0.2]$. If $\boldsymbol{Y}$ is mixed with the white Gaussian noise, leading to $\widetilde{\boldsymbol{Y}} = [0.0941, 0.2685; 0.1360, 0.2182]$, then $\widetilde{\boldsymbol{X}} = \boldsymbol{A}^{-1}\widetilde{\boldsymbol{Y}} = [-0.0521, -0.3189; 0.0731, 0.2937]$. However, if $y_{1,1}$ is corrupted by an anomaly, such that $\widetilde{\boldsymbol{Y}} = [3, 0.25; 0.15, 0.2]$, then $\widetilde{\boldsymbol{X}} = [-5.85, -0.3; 4.425, 0.275]$. That is, the Frobenius-norm-based solution deviates much from the ground truth.

One of the prevailing methods to resist anomalies is to adopt the ℓ_p-norm with $p \in (0, 2)$ as the loss function [39], leading to

$$\min_{\boldsymbol{\mathcal{X}}} \ \|\boldsymbol{\mathcal{X}}\|_{\mathrm{L}}, \quad \text{s.t. } \|\boldsymbol{\mathcal{X}} \odot \boldsymbol{\Omega} - \boldsymbol{\mathcal{M}}_{\Omega}\|_p \le \delta \tag{10}$$

where $\delta > 0$ is a tolerance parameter to control the fitting error, and $\|\cdot\|_{\mathrm{L}}$ is defined as the logarithmic norm; see [39].

Another strategy is to exploit RPCA, and the corresponding algorithms include tensor nuclear norm with total variation regularization (TNTV) [51] and transformed t-SVD (TTSVD) [52]. Specifically, TTSVD considers the following optimization problem:

$$\begin{aligned}&\min_{\mathcal{X},\mathcal{L},\mathcal{S}} \ \|\mathcal{L}\|_{\mathrm{TTNN}}+\mu\|\mathcal{S}\|_1\\ &\text{s.t. } \mathcal{L}+\mathcal{S}=\mathcal{X},\ \mathcal{X}\odot\mathbf{\Omega}=\mathcal{M}_\Omega\end{aligned} \tag{11}$$

where $\mu > 0$ is the penalty parameter, and $\|\cdot\|_{\mathrm{TTNN}}$ is the transformed tubal nuclear norm, as defined in [52]. Ideally, (11) is able to attain good performance in both the Gaussian noise and impulsive noise via tweaking μ. However, it is time-consuming to tune μ to attain satisfactory performance for different data and noise environments.

III. Proposed Algorithm

In this section, we first present the suggested method. Then, we analyze its convergence behavior and computational complexity.

A. Algorithm Development

To restrain gross errors, we suggest solving tensor recovery using the capped Frobenius norm, defined as follows:

$$\|\mathcal{X}\|_{CF}=\sqrt{\sum_{l_1,\ldots,l_N=1}^{I_1,\ldots,I_N}\min\left(x^2_{l_1,\ldots,l_N},\theta^2\right)} \tag{12}$$

where $\theta > 0$ is an upper bound to suppress anomalies. Note that the upper limit θ can be considered as the threshold for differentiating the normal and anomaly-contaminated elements. In this study, we adopt the normalized median absolute deviation for its adaptive determination, which is shortly introduced in (28).

Before proceeding, we highlight the advantages of the capped Frobenius norm over the Frobenius norm and ℓ_p-norm with $p \in (0, 2)$. Compared with the Frobenius norm, in impulsive noise, the capped Frobenius norm is able to restrain the outliers. When there are no outliers, it reverts to the Frobenius norm. In comparison to the ℓ_p-norm, the capped Frobenius norm resists gross errors, but it does not affect the normal entries. As a result, the capped Frobenius norm has a better performance than the ℓ_p-norm in the presence of impulsive noise, as well as achieves comparable performance to the Frobenius norm in the Gaussian noise scenarios.

We then combine the capped Frobenius norm with tensor ring decomposition to formulate the RTC problem as follows:

$$\min_{\mathcal{Y}_{n:n=1,\ldots,N}} \|\mathcal{M}_{\mathbf{\Omega}}-h(\mathcal{Y}_1\cdots\mathcal{Y}_N)\odot\mathbf{\Omega}\|^2_{CF} \tag{13}$$

where $\mathcal{Y}_n \in \mathbb{R}^{R\times I_n\times R}$ with R being the predefined multilinear tensor ring rank. Since the capped Frobenius norm is nonconvex, (13) w.r.t. each variable is nonconvex. We exploit the half-quadratic theory to convert (13) into a tractable problem, such that it is a convex optimization w.r.t. each individual variable. The relevant background is introduced in the following lemma.

Lemma 1: [46]: Given $\phi(y)$ and $\psi(x)$, if $\phi(y)$ makes $f(y) = y^2 - \phi(y)$ convex, and $\psi(x)$ generates a convex function $g(x) = x^2 + \psi(x)$, then we have

$$\phi(y)=\inf_x \ ((y-x)^2+\psi(x)),\quad y\in(-\infty,+\infty) \tag{14a}$$

$$\psi(x)=\sup_y \ (-(y-x)^2+\phi(y)),\quad x\in(-\infty,+\infty). \tag{14b}$$

Based on Lemma 1, we derive an equivalent convex function of $\phi_\theta(y) = \min\ (y^2, \theta^2)$, as described in Theorem 1.

Theorem 1: Let $\phi_\theta(y) = \min\ (y^2, \theta^2)$, then minimizing $\phi_\theta(y)$ is equivalent to

$$\min_{x,y} \ (y-x)^2+\psi_\theta(x) \tag{15}$$

where $\psi_\theta(x)$ is

$$\psi_\theta(x)=\begin{cases}-(\theta-|x|)^2+\theta^2, & |x|<\theta\\ \theta^2 & |x|\geq\theta.\end{cases} \tag{16}$$

The proof is provided in Appendix A. Note that all the appendices are presented in supplementary material.

Moreover, we introduce a set Φ involving the coordinates of the observed entries in $\mathcal{M}_{\mathbf{\Omega}}$, defined as follows:

$$\Phi=\{(l_1,\ldots,l_N)|\Omega_{l_1,\ldots,l_N}=1\}. \tag{17}$$

Based on Φ, (13) is reexpressed as follows:

$$\min_{\mathcal{Y}_{n:n=1,\ldots,N}}\sum_{(l_1,\ldots,l_N)\in\Phi}\phi_\theta(m_{l_1,\ldots,l_N}-h(\mathcal{Y}_1\cdots\mathcal{Y}_N)_{l_1,\ldots,l_N})^2. \tag{18}$$

In accordance to Theorem 1, (18) is equivalent to

$$\min_{\mathcal{Y}_{n:n=1,\ldots,N}}\sum_{(l_1,\ldots,l_N)\in\Phi}\Big((m_{l_1,\ldots,l_N}-h(\mathcal{Y}_1\cdots\mathcal{Y}_N)_{l_1,\ldots,l_N} -s_{l_1,\ldots,l_N})^2+\psi_\theta(s_{l_1,\ldots,l_N})\Big). \tag{19}$$

Defining $\psi_\theta(\mathcal{S}_{\mathbf{\Omega}}) = \sum_{(l_1,\ldots,l_N)\in\Phi}\psi_\theta(s_{l_1,\ldots,l_N})$, (19) is then reformulated as follows:

$$\min_{\mathcal{S},\mathcal{Y}_{n:n=1,\ldots,N}}\|\mathcal{M}_{\mathbf{\Omega}}-h(\mathcal{Y}_1\cdots\mathcal{Y}_N)\odot\mathbf{\Omega}-\mathcal{S}_{\mathbf{\Omega}}\|_F^2+\psi_\theta(\mathcal{S}_{\mathbf{\Omega}}). \tag{20}$$

It is clear that (20) is a multivariable nonconvex optimization problem, but w.r.t. each individual variable, it is convex. To tackle (20), PBCD is adopted as the solver, resulting in the following iterative procedure:

$$\begin{aligned}\mathcal{S}^{k+1}&=\arg\min_{\mathcal{S}}\left\|\mathcal{M}_{\mathbf{\Omega}}-h\left(\mathcal{Y}_1^k\mathcal{Y}_2^k\cdots\mathcal{Y}_N^k\right)\odot\mathbf{\Omega}-\mathcal{S}_{\mathbf{\Omega}}\right\|_F^2\\ &\quad+\psi_\theta(\mathcal{S}_{\mathbf{\Omega}})\end{aligned} \tag{21a}$$

$$\begin{aligned}\mathcal{Y}_1^{k+1}&=\arg\min_{\mathcal{Y}_1}\left\|\mathcal{M}_{\mathbf{\Omega}}-h\left(\mathcal{Y}_1\mathcal{Y}_2^k\cdots\mathcal{Y}_N^k\right)\odot\mathbf{\Omega}-\mathcal{S}_{\mathbf{\Omega}}^{k+1}\right\|_F^2\\ &\quad+\lambda\left\|\mathcal{Y}_1-\mathcal{Y}_1^k\right\|_F^2\end{aligned} \tag{21b}$$

$$\begin{aligned}\mathcal{Y}_2^{k+1}&=\arg\min_{\mathcal{Y}_2}\left\|\mathcal{M}_{\mathbf{\Omega}}-h\left(\mathcal{Y}_1^{k+1}\mathcal{Y}_2\cdots\mathcal{Y}_N^k\right)\odot\mathbf{\Omega}-\mathcal{S}_{\mathbf{\Omega}}^{k+1}\right\|_F^2\\ &\quad+\lambda\left\|\mathcal{Y}_2-\mathcal{Y}_2^k\right\|_F^2\end{aligned} \tag{21c}$$

$$\vdots$$

$$\begin{aligned}\mathcal{Y}_N^{k+1}&=\arg\min_{\mathcal{Y}_N}\left\|\mathcal{M}_{\mathbf{\Omega}}-h\left(\mathcal{Y}_1^{k+1}\mathcal{Y}_2^{k+1}\cdots\mathcal{Y}_N\right)\odot\mathbf{\Omega}-\mathcal{S}_{\mathbf{\Omega}}^{k+1}\right\|_F^2\\ &\quad+\lambda\left\|\mathcal{Y}_N-\mathcal{Y}_N^k\right\|_F^2\end{aligned} \tag{21d}$$

where $\lambda > 0$ is the proximal parameter. It is seen that the PBCD alternately updates one of the variables with fixing the N remaining variables at each iteration.

We first tackle (21a). It is easy to know that its solution is only determined by the entries with $(l_1, \ldots, l_N) \in \Phi$. We thus reformulate (21a) as a vector optimization problem

$$\boldsymbol{s}^{k+1} = \arg\min_{\boldsymbol{s}} \|\boldsymbol{r}^k - \boldsymbol{s}\|_2^2 + \psi_\theta(\boldsymbol{s}) \tag{22}$$

where $\boldsymbol{r}^k \in \mathbb{R}^{\|\boldsymbol{\Omega}\|_1}$ consists of the observed entries of $\boldsymbol{\mathcal{R}}_{\boldsymbol{\Omega}}^k = \boldsymbol{\mathcal{M}}_{\boldsymbol{\Omega}} - h(\boldsymbol{\mathcal{Y}}_1^k \boldsymbol{\mathcal{Y}}_2^k \cdots \boldsymbol{\mathcal{Y}}_N^k) \odot \boldsymbol{\Omega}$. The procedure for obtaining $\boldsymbol{r}$ from $\boldsymbol{\mathcal{R}}_{\boldsymbol{\Omega}}^k$ is illustrated with the use of a third-order tensor as follows. Given $\boldsymbol{\Omega} \in \mathbb{R}^{2\times3\times2}$, such that

$$\boldsymbol{\Omega}_{:,:,1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \text{ and } \boldsymbol{\Omega}_{:,:,2} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{23a}$$

Then, $\boldsymbol{r}^k = [r_{2,1,1}^k, r_{1,2,1}^k, r_{2,3,1}^k, r_{1,1,2}^k, r_{2,2,2}^k, r_{1,3,2}^k]$ with $r_{i,j,k}^k$ being the (i, j, k) entry of $\boldsymbol{\mathcal{R}}_{\boldsymbol{\Omega}}^k$.

We then derive the closed-form solution to (22), describing in the following lemma.

Lemma 2: For the following optimization problem:

$$s^{k+1} = \arg\min_s \varphi(s) = \arg\min_s \ (r - s)^2 + \psi_\theta(s). \tag{24}$$

Its optimal solution is $T_\theta(r)$, defined as follows:

$$s^{k+1} = T_\theta(r) = \begin{cases} 0, & |r| < \theta \\ r, & |r| \geq \theta. \end{cases} \tag{25}$$

Besides, the subgradient of $\varphi(s)$ at minimizer s^{k+1} is

$$\partial\varphi\left(s^{k+1}\right) = \begin{cases} 0 \in [-(r+\theta), \theta - r], & |r| < \theta \\ 0, & |r| \geq \theta. \end{cases} \tag{26}$$

The proof is provided in Appendix B.

In (22), s_i^{k+1} only depends on r_i^k, and hence its optimal solution is

$$\boldsymbol{s}^{k+1} = T_{\theta^k}(\boldsymbol{r}^k). \tag{27}$$

Note that $\boldsymbol{s}^{k+1}$ is affected by the parameter θ^k. We suggest updating θ^k prior to computing $\boldsymbol{s}^{k+1}$ for better performance. Since $\boldsymbol{r}^k$ is defined as the fitting error at the kth iteration, if the mean of the fitting error is assumed 0, $-\theta < r < \theta$ is considered as a confidence interval to identify anomalies. To guarantee the convergence, it requires θ^k to be nonincreasing

$$\theta^k = \min(\hat{\theta}, \theta^{k-1}) \tag{28}$$

where $\hat{\theta}$ is determined by a robust measure for standard deviation, namely, the normalized median absolute deviation method [33]

$$\hat{\theta} = \zeta \times 1.4826 \times \mathrm{Med}(|\boldsymbol{r}^k - \mathrm{Med}(\boldsymbol{r}^k)|). \tag{29}$$

Here, $\zeta > 0$ controls the confidence interval range, and $\mathrm{Med}(\cdot)$ is the sample median operator. In Section IV, we will investigate the impact of ζ on the recovery performance in the Gaussian noise and impulsive noise scenarios.

After obtaining $\boldsymbol{s}^{k+1}$, $\boldsymbol{\mathcal{S}}^{k+1}$ is updated via the inverse operation of constructing $\boldsymbol{r}^k$ from $\boldsymbol{\mathcal{R}}_{\boldsymbol{\Omega}}^k$.

We then handle (21b)–(21d). Since they have the same structure, we detail the derivation procedure for one of them, say $\boldsymbol{\mathcal{Y}}_n^{k+1}$, without loss of generality. To simplify expressions, the optimization problem for updating $\boldsymbol{\mathcal{Y}}_n^{k+1}$ is reexpressed as follows:

$$\boldsymbol{\mathcal{Y}}_n^{k+1} = \arg\min_{\boldsymbol{\mathcal{Y}}_n} \left\|\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}} - h\left(\boldsymbol{\mathcal{Y}}_1^{k+1} \cdots \boldsymbol{\mathcal{Y}}_{n-1}^{k+1} \boldsymbol{\mathcal{Y}}_n \boldsymbol{\mathcal{Y}}_{n+1}^k \cdots \boldsymbol{\mathcal{Y}}_N^k\right) \odot \boldsymbol{\Omega}\right\|_F^2 + \lambda\left\|\boldsymbol{\mathcal{Y}}_n - \boldsymbol{\mathcal{Y}}_n^k\right\|_F^2 \tag{30}$$

where $\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}} = \boldsymbol{\mathcal{M}}_{\boldsymbol{\Omega}} - \boldsymbol{\mathcal{S}}_{\boldsymbol{\Omega}}^{k+1}$. Using tensor permutation operation, (30) is equivalent to

$$\boldsymbol{\mathcal{Y}}_n^{k+1} = \arg\min_{\boldsymbol{\mathcal{Y}}_n} \left\|\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}}^{T_n} - h\left(\boldsymbol{\mathcal{Y}}_n \boldsymbol{\mathcal{Y}}\right) \odot \boldsymbol{\Omega}^{T_n}\right\|_F^2 + \lambda\left\|\boldsymbol{\mathcal{Y}}_n - \boldsymbol{\mathcal{Y}}_n^k\right\|_F^2 \tag{31}$$

where $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{Y}}_{n+1}^k \cdots \boldsymbol{\mathcal{Y}}_N^k \boldsymbol{\mathcal{Y}}_1^{k+1} \cdots \boldsymbol{\mathcal{Y}}_{n-1}^{k+1}$ with the dimensions of $R \times (I_{n+1} \cdots I_N I_1 \cdots I_{n-1}) \times R$. We further adopt mode-$n$ unfolding to recast (31) as the following matrix optimization problem:

$$\boldsymbol{\mathcal{Y}}_n^{k+1} = \arg\min_{\boldsymbol{\mathcal{Y}}_n} \left\|\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}\,[n]}^{T_n} - h\left(\boldsymbol{\mathcal{Y}}_n \boldsymbol{\mathcal{Y}}\right)_{[n]} \odot \boldsymbol{\Omega}_{[n]}^{T_n}\right\|_F^2 + \lambda\left\|\boldsymbol{\mathcal{Y}}_n - \boldsymbol{\mathcal{Y}}_n^k\right\|_F^2 \tag{32}$$

where the dimensions of $\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}\,[n]}^{T_n}$, $h(\boldsymbol{\mathcal{Y}}_n\boldsymbol{\mathcal{Y}})_{[n]}$, and $\boldsymbol{\Omega}_{[n]}^{T_n}$ are $I_n \times (I_1 \cdots I_{n-1} I_{n+1} \cdots I_N)$. Since the lth lateral slice of $\boldsymbol{\mathcal{Y}}_n$, denoted as $\boldsymbol{\mathcal{Y}}_n(:, l, :)$, corresponds to the lth row of $h(\boldsymbol{\mathcal{Y}}_n\boldsymbol{\mathcal{Y}})_{[n]}$, (32) is split into I_n subproblems

$$\begin{aligned} &\boldsymbol{\mathcal{Y}}_n^{k+1}(:, l, :) \\ &= \arg\min_{\boldsymbol{\mathcal{Y}}_n} \left\|\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}\,[n]}^{T_n}(l, :) - h(\boldsymbol{\mathcal{Y}}_n(:, l, :)\boldsymbol{\mathcal{Y}})_{[n]} \odot \boldsymbol{\Omega}_{[n]}^{T_n}(l, :)\right\|_F^2 \\ &\quad + \lambda\left\|\boldsymbol{\mathcal{Y}}_n(:, l, :) - \boldsymbol{\mathcal{Y}}_n^k(:, l, :)\right\|_F^2, \text{ for } l \in [1, I_n] \end{aligned} \tag{33}$$

where the lengths of $\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}\,[n]}^{T_n}(l, :)$, $h(\boldsymbol{\mathcal{Y}}_n(:, l, :)\boldsymbol{\mathcal{Y}})_{[n]}$, and $\boldsymbol{\Omega}_{[n]}^{T_n}(l, :)$ are $I_1 \cdots I_{n-1} I_{n+1} \cdots I_N$. Equation (33) indicates that $\boldsymbol{\mathcal{Y}}_n$ can be updated in a distributed or parallel manner. Similar to the update of $\boldsymbol{\mathcal{S}}$, the solution to (33) is only affected by the observed entries, and thus (33) is equivalent to

$$\boldsymbol{\mathcal{Y}}_n^{k+1}(:, l, :) = \arg\min_{\boldsymbol{\mathcal{Y}}_n} \|\boldsymbol{g}_{\boldsymbol{\Omega}_l} - h(\boldsymbol{\mathcal{Y}}_n(:, l, :)\boldsymbol{\mathcal{Y}}_{\boldsymbol{\Omega}_l})_{[n]}\|_F^2 + \lambda\left\|\boldsymbol{\mathcal{Y}}_n(:, l, :) - \boldsymbol{\mathcal{Y}}_n^k(:, l, :)\right\|_F^2 \tag{34}$$

where $\boldsymbol{g}_{\boldsymbol{\Omega}_l} \in \mathbb{R}^{\|\boldsymbol{\Omega}_l\|_1}$ and $\boldsymbol{\mathcal{Y}}_{\boldsymbol{\Omega}_l} \in \mathbb{R}^{R\times\|\boldsymbol{\Omega}_l\|_1\times R}$ consist of the observed elements of $\boldsymbol{\mathcal{G}}_{\boldsymbol{\Omega}\,[n]}^{T_n}(l, :)$ and $\boldsymbol{\mathcal{Y}}$, respectively. Herein, $\boldsymbol{\Omega}_l$ only includes entries of 1 in $\boldsymbol{\Omega}_{[n]}^{T_n}(l, :)$. We represent the tensor $\boldsymbol{\mathcal{Y}}_n(:, l, :) \in \mathbb{R}^{R\times1\times R}$ by a matrix $\boldsymbol{Y}_n \in \mathbb{R}^{R\times R}$. Based on the elementwise expression of tensor ring, (34) is reformulated as follows:

$$\boldsymbol{\mathcal{Y}}_n^{k+1}(:, l, :) = \arg\min_{\boldsymbol{Y}_n} \sum_{j=1}^{\|\boldsymbol{\Omega}_l\|_1} \left(\boldsymbol{g}_{\boldsymbol{\Omega}_l}(j) - \mathrm{tr}(\boldsymbol{Y}_n \times \boldsymbol{\mathcal{Y}}_{\boldsymbol{\Omega}_l}(:, j, :))\right)^2 + \lambda\left\|\boldsymbol{Y}_n - \boldsymbol{Y}_n^k\right\|_F^2. \tag{35}$$

To handle (35), we introduce the following lemma.

Lemma 3: [29]: Consider $\boldsymbol{U} \in \mathbb{R}^{I_1\times I_2}$ and $\boldsymbol{V} \in \mathbb{R}^{I_2\times I_1}$, we have

$$\mathrm{tr}(\boldsymbol{U} \times \boldsymbol{V}) = \mathrm{vec}(\boldsymbol{V}^T)^T \mathrm{vec}(\boldsymbol{U}). \tag{36}$$

Algorithm 1 CFN-RTC

Input: Partially observed tensor $\mathcal{M}_{\mathbf{\Omega}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, binary tensor $\mathbf{\Omega} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, tensor ring rank R, maximum iteration number $K_{\max}$, and proximal parameter λ

Initialize: Randomize $\mathcal{Y}_n^1 \in \mathbb{R}^{R \times I_n \times R}$ with $n \in [1, N]$ and $\theta^0 = 10$

for $k = 1, 2, \cdots, K_{\max}$ **do**
 1) Compute θ^k via (28)
 2) Compute $\boldsymbol{s}^{k+1}$ via (27)
 3) Update $\mathcal{S}^{k+1}$ based on $\boldsymbol{s}^{k+1}$
 for $n = 1 : N$ **do**
  for $l = 1 : I_n$ **do**
   4) Compute $\boldsymbol{y}_n^{k+1}$ via (40)
   5) Update $\mathcal{Y}_n^{k+1}(:, l, :)$ based on $\boldsymbol{y}_n^{k+1}$
  end for
 end for
 Stop if stopping criterion is met.
end for

Output: $\mathcal{X} = h(\mathcal{Y}_1^{k+1}\mathcal{Y}_2^{k+1} \cdots \mathcal{Y}_N^{k+1})$

Then, in accordance to Lemma 3, (35) is rewritten as follows:

$$\mathcal{Y}_n^{k+1}(:, l, :) = \arg\min_{\boldsymbol{Y}_n} \left\|\boldsymbol{g}_{\mathbf{\Omega}_l} - \boldsymbol{A}_l \mathrm{vec}(\boldsymbol{Y}_n)\right\|_F^2 + \lambda \left\|\boldsymbol{Y}_n - \boldsymbol{Y}_n^k\right\|_F^2 \tag{37}$$

where $\boldsymbol{A}_l \in \mathbb{R}^{\|\mathbf{\Omega}_l\|_1 \times R^2}$ and $\boldsymbol{A}_l(j, :) = \mathrm{vec}(\mathcal{Y}_{\mathbf{\Omega}_l}(:, j, :)^T)^T$ with $j \in [1, \|\mathbf{\Omega}_l\|_1]$. Note that the first term corresponds to a vector optimization problem w.r.t. $\boldsymbol{Y}_n$, while the second term is matrix optimization. To be consistent, we reformulate (37) as follows:

$$\boldsymbol{y}_n^{k+1} = \arg\min_{\boldsymbol{y}_n} \left\|\boldsymbol{A}_l \boldsymbol{y}_n - \boldsymbol{g}_{\mathbf{\Omega}_l}\right\|_F^2 + \lambda \left\|\boldsymbol{y}_n - \boldsymbol{y}_n^k\right\|_F^2 \tag{38}$$

where $\boldsymbol{y}_n = \mathrm{vec}(\boldsymbol{Y}_n) \in \mathbb{R}^{R^2}$. Since (38) is a quadratic problem, its optimal solution can be obtained by the following equation:

$$2\boldsymbol{A}_l^T\left(\boldsymbol{A}_l \boldsymbol{y}_n^{k+1} - \boldsymbol{g}_{\mathbf{\Omega}_l}\right) + 2\lambda\left(\boldsymbol{y}_n^{k+1} - \boldsymbol{y}_n^k\right) = \mathbf{0} \tag{39}$$

resulting in the solution

$$\boldsymbol{y}_n^{k+1} = \left(\boldsymbol{A}_l^T \boldsymbol{A}_l + \lambda \boldsymbol{I}\right)^{-1}\left(\boldsymbol{A}_l^T \boldsymbol{g}_{\mathbf{\Omega}_l} + \lambda \boldsymbol{y}_n^k\right). \tag{40}$$

After obtaining $\boldsymbol{y}_n^{k+1}$, the solution to (37) is determined as follows:

$$\mathcal{Y}_n^{k+1}(:, l, :) = \boldsymbol{Y}_n^{k+1} = \mathrm{mat}\left(\boldsymbol{y}_n^{k+1}\right). \tag{41}$$

The procedure for updating $\mathcal{Y}_n^{k+1}$ is complete. We name the proposed algorithm capped Frobenius-norm-based RTC (CFN-RTC). Its steps are summarized in Algorithm 1. The CFN-RTC has two stopping criteria. One is to reach the maximum iteration number $K_{\max}$. Our experiments suggest that $K_{\max} = 50$ is sufficient to ensure convergence. The other one depends on a tolerance parameter, defined as follows:

$$\eta = \frac{\left\|h\left(\mathcal{Y}_1^{k+1}\mathcal{Y}_2^{k+1} \cdots \mathcal{Y}_N^{k+1}\right) - h\left(\mathcal{Y}_1^k \mathcal{Y}_2^k \cdots \mathcal{Y}_N^k\right)\right\|_F^2}{\left\|h\left(\mathcal{Y}_1^k \mathcal{Y}_2^k \cdots \mathcal{Y}_N^k\right)\right\|_F^2}. \tag{42}$$

In our study, when $\eta < 10^{-4}$ is reached, the algorithm will terminate.

B. Comparison With RPCA

From (20), we see that the capped Frobenius-norm-based formulation is converted into a form similar to RPCA. The main difference lies on the regularization term, that is, the ℓ_1-norm in prevailing RPCA and $\psi_\theta(\cdot)$ in our method. The convexity of the ℓ_1-norm generates a tractable optimization; however, it may overpenalize large components, which causes the solution to deviate from the ground truth. Although our $\psi_\theta(\cdot)$ is nonconvex, the resultant subproblem is convex and has a closed-form solution. On the other hand, the performance of RPCA and our method are affected by an auxiliary parameter. To our best knowledge, the tradeoff parameter in RPCA requires tweaking manually to attain good performance. While for the proposed algorithm, θ is automatically updated using a robust statistics-based method.

C. Convergence Analysis

In this section, we analyze the convergence behavior of CFN-RTC. To facilitate presentation, the analysis is based on a third-order tensor, that is, $\mathcal{X} = h(\mathcal{Y}_1\mathcal{Y}_2\mathcal{Y}_3) \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. It is worth mentioning that the analysis can be extended to higher order tensors. We first define the objective function value as follows:

$$\begin{aligned}&\mathcal{L}_{\theta^k}\left(\mathcal{S}_{\mathbf{\Omega}}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k\right)\\&\quad = \left\|\mathcal{M}_{\mathbf{\Omega}} - h\left(\mathcal{Y}_1^k\mathcal{Y}_2^k\mathcal{Y}_3^k\right) \odot \mathbf{\Omega} - \mathcal{S}_{\mathbf{\Omega}}^k\right\|_F^2 + \psi_{\theta^k}\left(\mathcal{S}_{\mathbf{\Omega}}^k\right).\end{aligned} \tag{43}$$

The convergence behavior of $\mathcal{L}_{\theta^k}(\mathcal{S}_{\mathbf{\Omega}}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)$ is provided in Theorem 2.

Theorem 2: Let $\mathcal{L}_{\theta^k}(\mathcal{S}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)$ be the objective function value generated by Algorithm 1, then we have the following statements.

1) $\mathcal{L}_{\theta^k}(\mathcal{S}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)$ is nonincreasing with all the variables' update.
2) $\mathcal{L}_{\theta^k}(\mathcal{S}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)$ is lower bounded.

Therefore, $\{\mathcal{L}_{\theta^k}(\mathcal{S}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)\}_{k\in\mathbb{N}}$ is convergent.

The proof is provided in Appendix C.

We then analyze the sequence behavior in Theorem 3. The definition of the critical point is first introduced using the following lemma.

Lemma 4: [53]: Given a function $\varphi(x)$, then x^* is a critical point if x^* meets one of the following statements.

1) $\nabla\varphi(x^*) = 0$ in the case of smooth $\varphi(x)$.
2) $0 \in \partial\varphi(x^*)$ where $\partial\varphi(x^*)$ is the subgradient with the nonsmooth $\varphi(x)$.

Theorem 3: Let $\{(\mathcal{S}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)\}$ be the sequence generated by Algorithm 1. For any initialization with finite $\mathcal{L}_{\theta^1}(\mathcal{S}^1, \mathcal{Y}_1^1, \mathcal{Y}_2^1, \mathcal{Y}_3^1)$, $\{(\mathcal{S}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)\}$ meets the following properties.

1) The sequence $\{(\mathcal{S}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)\}$ is bounded.
2) There exists a subsequence $\{(\mathcal{S}^{k_i}, \mathcal{Y}_1^{k_i}, \mathcal{Y}_2^{k_i}, \mathcal{Y}_3^{k_i})\}$ converging to an accumulation point $(\mathcal{S}^*, \mathcal{Y}_1^*, \mathcal{Y}_2^*, \mathcal{Y}_3^*)$.
3) The accumulation point $(\mathcal{S}^*, \mathcal{Y}_1^*, \mathcal{Y}_2^*, \mathcal{Y}_3^*)$ is a critical point.

The proof is provided in Appendix D.

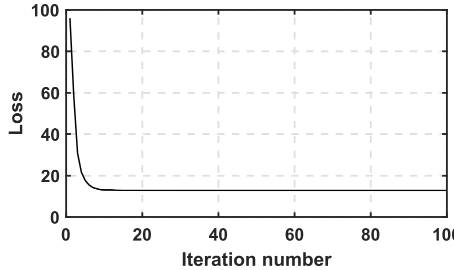

Fig. 1. Convergence behavior of the objective function value with 50% randomly missing data and 3-dB GMM noise.

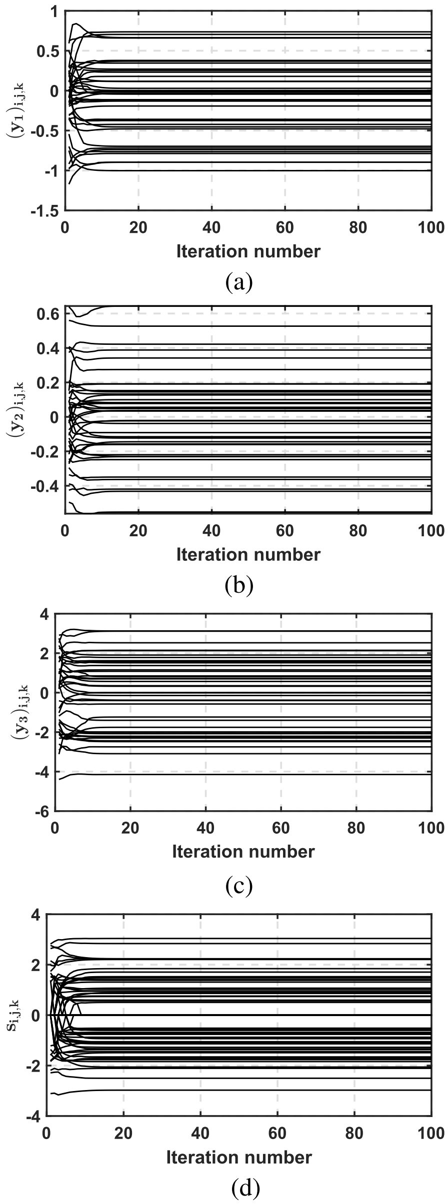

Fig. 2. Sequence convergence behavior of CFN-RTC with 50% randomly missing data and 3-dB GMM noise. (a) Convergence of elements in $\mathcal{Y}_1$. (b) Convergence of elements in $\mathcal{Y}_2$. (c) Convergence of elements in $\mathcal{Y}_3$. (d) Convergence of elements in $\mathcal{S}$.

D. Complexity Analysis

Here, the analysis is also based on a third-order tensor, that is, $\mathcal{X} = h(\mathcal{Y}_1\mathcal{Y}_2\mathcal{Y}_3)$ with $\mathcal{Y}_1 \in \mathbb{R}^{R\times I_1\times R}$, $\mathcal{Y}_2 \in \mathbb{R}^{R\times I_2\times R}$, and $\mathcal{Y}_3 \in \mathbb{R}^{R\times I_3\times R}$.

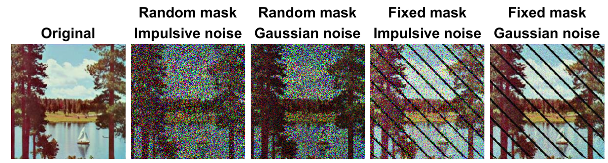

Fig. 3. Scenery and its corrupted versions.

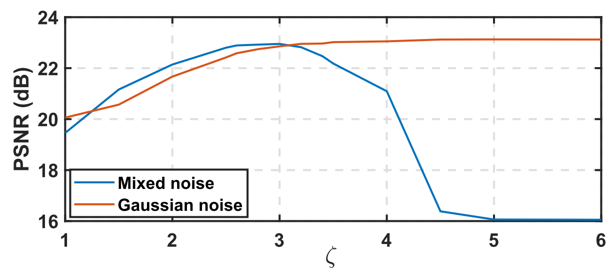

Fig. 4. PSNR versus ζ.

For the update of $\mathcal{S}$, the computational complexity is dominated by the calculation of $h(\mathcal{Y}_1\mathcal{Y}_2\mathcal{Y}_3)$. One efficient method is to compute the observed entries, resulting in the complexity of $\mathcal{O}(pI_1I_2I_3R^3)$ where p is the observation ratio. In addition, the complexity to update θ_k is $\mathcal{O}(pI_1I_2I_3)$. For updating $\mathcal{Y}_n^k(:,l,:)$ with $n = 1,2,3$ by (38), the complexity is $\mathcal{O}(\|\mathbf{\Omega}_l\|_1 R^4)$. Therefore, the complexity to update $\mathcal{Y}_n^k$ is $\mathcal{O}(pI_1I_2I_3R^4)$ due to $pI_1I_2I_3 = \sum_{l=1}^{I_n}\|\mathbf{\Omega}_l\|_1$. As a result, the total computational complexity is $\mathcal{O}(pI_1I_2I_3R^4)$ per iteration.

IV. Experimental Results

In this section, we evaluate the CFN-RTC using synthetic data, real-world images, and videos. Note that λ is set to 10^{-8} in all the experiments. The competing methods include tensor ring completion (TRC) [29], ℓ_p-norm based tensor train completion (ℓ_p-TTC) [37], LNOP [39], TNTV [51], TTSVD [52], and RTC with rank estimation (RTC-RE) [54].

A. Convergence Behavior

We first verify the convergence behavior of the suggested method based on a small-size synthetic data, i.e., $\mathcal{M} \in \mathbb{R}^{10\times 10\times 10}$. The complete tensor is generated by three tensor ring factors, namely, $\mathcal{M}_1 \in \mathbb{R}^{2\times 10\times 2}$, $\mathcal{M}_2 \in \mathbb{R}^{2\times 10\times 2}$, and $\mathcal{M}_3 \in \mathbb{R}^{2\times 10\times 2}$ whose entries obey the standard Gaussian distribution. Then, the incomplete noise-free tensor $\mathcal{M}_{\mathbf{\Omega}}$ consists of randomly selected 50% entries of $\mathcal{M}$. Moreover, $\widetilde{\mathcal{M}}_{\mathbf{\Omega}}$ is contaminated with independent impulsive noise which is modeled by a Gaussian mixture model (GMM). The probability density function (PDF) of GMM is given by the following equation:

$$p_v(v) = \frac{c_1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{v^2}{2\sigma_1^2}\right) + \frac{c_2}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{v^2}{2\sigma_2^2}\right) \quad (44)$$

where $c_1 + c_2 = 1$ with $0 < c_i < 1$, and σ_1^2 and σ_2^2 are the variances. To simulate the impulsive noise, we take $\sigma_2^2 \gg \sigma_1^2$ and $c_2 < c_1$. This means that sparse and high power noise

Fig. 5. Recovered Scenery images by different algorithms. The first and second rows contain the results with the random mask with impulsive noise and Gaussian noise, respectively. The third row shows the restored images with the deterministic mask and impulsive noise, while the fourth row contains the reconstructed pictures with deterministic mask and Gaussian noise.

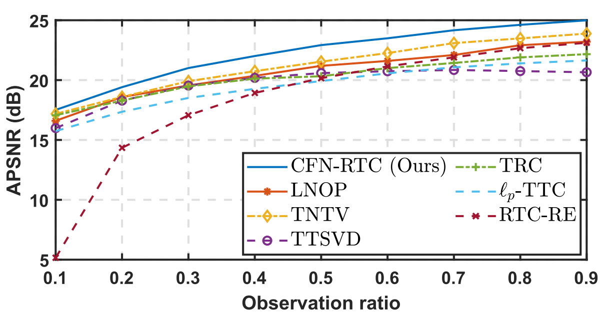

Fig. 6. Average PSNR (APSNR) versus observation ratio.

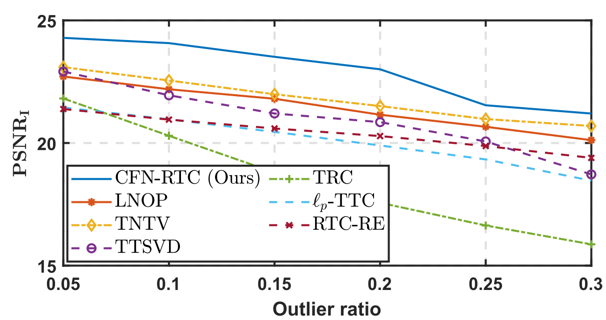

Fig. 7. PSNR_I versus outlier ratio.

samples corresponding to σ_2^2 and c_2 are mixed in Gaussian background noise with small variance σ_1^2. We set $\sigma_2^2 = 100\sigma_1^2$ and $c_2 = 0.1$. The signal-to-noise ratio (SNR) is defined as follows:

$$\mathrm{SNR} = \frac{\|\mathcal{M}_{\mathbf{\Omega}}\|_F^2}{\|\mathbf{\Omega}\|_1 \sigma_v^2} \tag{45}$$

where $\sigma_v^2 = \sum_{i=1}^{2} c_i \sigma_i^2$ is the total noise variance.

The convergence of the objective function value is investigated in Fig. 1. It is seen that the objective function value is nonincreasing and converges within 20 iterations.

In addition, Fig. 2 depicts the sequence convergence behavior of four variables, namely, $\mathcal{Y}_1 \in \mathbb{R}^{2\times 10\times 2}$, $\mathcal{Y}_2 \in \mathbb{R}^{2\times 10\times 2}$, $\mathcal{Y}_3 \in \mathbb{R}^{2\times 10\times 2}$, and $\mathcal{S} \in \mathbb{R}^{10\times 10\times 10}$. It is noted that the number of curves for $\mathcal{S}$ is much less than 1000 because of its sparsity. We have already proved analytically the subsequence convergence of $\{(\mathcal{S}^{k_i}, \mathcal{Y}_1^{k_i}, \mathcal{Y}_2^{k_i}, \mathcal{Y}_3^{k_i})\}$, and the simulations corroborate our theory, showing that convergence happens within 20 iterations in this case.

Fig. 8. Eight images.

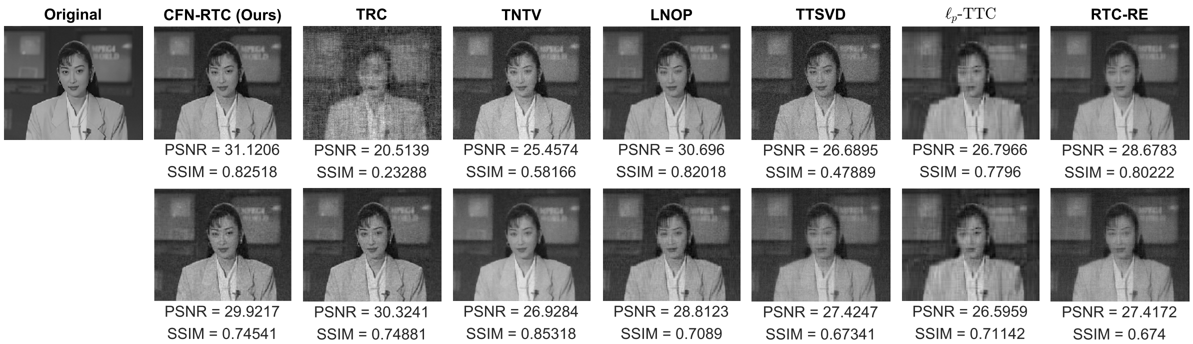

Fig. 9. Original and recovered frames for Akiyo. Top row corresponds to impulsive noise, while bottom row corresponds to Gaussian noise.

TABLE II

PERFORMANCE BY DIFFERENT ALGORITHMS ON EIGHT IMAGES WITH RANDOM MASK

		Image-1	Image-2	Image-3	Image-4	Image-5	Image-6	Image-7	Image-8
CFN-RTC (Ours)	$\mathrm{PSNR_I}$	24.346	27.656	24.091	29.841	28.507	24.928	27.626	23.875
	$\mathrm{Time_I}$	36.541	20.953	33.016	26.003	21.030	33.431	37.542	32.819
	$\mathrm{PSNR_G}$	23.778	26.409	23.625	26.300	26.554	24.198	26.037	23.637
	$\mathrm{Time_G}$	33.118	20.474	32.825	25.719	17.228	33.017	37.126	34.814
	APSNR	**24.062**	**27.032**	**23.858**	**28.071**	**27.531**	**24.563**	**26.832**	**23.756**
TRC [29]	$\mathrm{PSNR_I}$	16.803	18.956	16.293	17.752	19.234	16.035	16.382	16.235
	$\mathrm{Time_I}$	29.607	13.305	28.527	19.131	13.883	27.904	20.096	27.771
	$\mathrm{PSNR_G}$	24.112	26.821	23.925	26.584	26.789	24.566	26.261	24.121
	$\mathrm{Time_G}$	27.431	14.952	28.394	18.731	13.274	27.896	19.189	27.451
	APSNR	20.456	22.889	20.109	22.168	23.012	20.301	21.322	20.178
LNOP [39]	$\mathrm{PSNR_I}$	23.983	24.171	22.596	27.181	27.774	21.795	25.706	22.082
	$\mathrm{Time_I}$	6.419	6.889	7.222	6.536	6.620	6.479	6.485	6.592
	$\mathrm{PSNR_G}$	23.154	23.238	21.955	25.709	26.223	21.271	24.647	21.861
	$\mathrm{Time_G}$	6.411	6.445	6.743	6.433	6.500	6.575	6.333	6.440
	APSNR	23.569	23.705	22.276	26.445	26.999	21.533	25.177	21.972
TTSVD [52]	$\mathrm{PSNR_I}$	23.628	23.959	21.987	25.135	25.082	21.925	24.926	22.096
	$\mathrm{Time_I}$	17.913	18.092	14.948	20.095	19.600	15.433	22.079	16.992
	$\mathrm{PSNR_G}$	21.412	21.385	20.625	21.953	21.902	20.568	22.071	20.829
	$\mathrm{Time_G}$	13.815	13.535	13.103	14.121	13.985	13.696	15.965	13.582
	APSNR	22.520	22.672	21.306	23.544	23.492	21.247	23.499	21.463
TNTV [51]	$\mathrm{PSNR_I}$	22.025	22.129	21.314	27.596	24.388	17.149	27.357	23.379
	$\mathrm{Time_I}$	16.369	16.285	16.365	16.205	16.471	16.048	16.595	16.156
	$\mathrm{PSNR_G}$	22.281	22.171	21.963	26.611	23.696	19.425	26.232	23.392
	$\mathrm{Time_G}$	16.351	16.403	16.079	16.376	16.079	15.873	16.181	15.991
	APSNR	22.153	22.150	21.639	27.104	24.042	19.425	26.795	23.386
ℓ_p-TTC [37]	$\mathrm{PSNR_I}$	23.126	23.514	21.536	26.564	24.209	21.059	24.732	21.025
	$\mathrm{Time_I}$	95.562	89.312	99.606	91.808	90.626	96.618	90.495	98.261
	$\mathrm{PSNR_G}$	21.834	21.681	20.865	23.972	23.011	20.451	22.834	20.888
	$\mathrm{Time_G}$	93.475	90.206	96.056	97.186	97.283	90.393	95.145	89.298
	APSNR	22.480	22.598	21.2001	25.268	23.610	20.755	23.783	20.957
RTC-RE [54]	$\mathrm{PSNR_I}$	22.849	23.109	21.636	26.247	27.215	20.718	24.474	21.309
	$\mathrm{Time_I}$	42.797	37.532	41.957	41.873	34.929	41.479	39.143	40.474
	$\mathrm{PSNR_G}$	22.554	22.665	21.393	25.003	25.562	20.668	23.919	21.438
	$\mathrm{Time_G}$	40.359	37.448	39.525	36.450	26.067	40.735	37.871	42.897
	APSNR	22.701	22.887	21.514	25.625	26.389	20.693	23.851	21.374

B. Image Inpainting

One popular application of TC is color image inpainting [55]. Color images involve RGB channels, and one channel can be modeled as a matrix. Therefore, color images can be represented by the third-order tensors. In practice, images may not be entirely acquired owing to the damage to the photosensitive device or shadow cast by other objects. Furthermore, images may be corrupted by the white Gaussian noise or impulsive noise during wireless transmission or bit errors in the signal acquisition stage. In the following experiments, we consider two types of noise, namely, strong Gaussian noise with $\sigma^2 = 0.01$ as well as impulsive noise generated by the mixture of weak white Gaussian noise with $\sigma^2 = 0.002$ and salt-and-pepper noise with $\tau = 0.2$ where σ^2 and τ are the variance and density coefficient, respectively.

The examined image is Scenery with dimensions of $256 \times 256 \times 3$ [56]. Besides, we investigate two types of masks, namely, random and fixed masks [57]. The random mask implies that the image has randomly distributed missing pixels, while the deterministic mask corresponds to regular stripes. Fig. 3 depicts the original Scenery and four corrupted versions, i.e., random mask with Gaussian noise, random mask with impulsive noise, fixed mask with Gaussian noise, and fixed mask with impulsive noise. To evaluate recovery performance, two widely used metrics are adopted, namely, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). Note that large PSNR and SSIM indicate good restoration performance.

We first investigate the impact of ζ in (29) on recovery performance. The results are plotted in Fig. 4 where the incomplete image has 50% randomly missing pixels. We see that the PSNR with Gaussian noise increases with ζ, while PSNR, in impulsive noise scenarios, first increases and then reduces with boosting the value of ζ. This is because a smaller ζ results in a narrow confidence interval, indicating that more entries are considered as the outlier-contaminated

TABLE III
PERFORMANCE BY DIFFERENT ALGORITHMS ON EIGHT IMAGES WITH DETERMINISTIC MASK

		Image-1	Image-2	Image-3	Image-4	Image-5	Image-6	Image-7	Image-8
CFN-RTC (Ours)	PSNR_I	25.709	27.739	24.992	30.521	28.813	25.818	28.927	24.363
	Time_I	48.004	25.606	43.112	32.849	37.388	42.130	32.918	40.041
	PSNR_G	24.321	27.231	24.536	28.146	27.591	25.135	27.581	24.442
	Time_G	43.782	25.381	43.146	33.061	37.102	43.248	32.807	39.689
	APSNR	**25.015**	**27.485**	**24.764**	**29.334**	**28.202**	**25.227**	**28.254**	**24.403**
TRC [29]	PSNR_I	17.809	20.077	18.523	20.161	21.532	18.234	18.563	17.873
	Time_I	37.452	20.922	30.012	29.166	21.598	35.521	25.561	34.812
	PSNR_G	24.531	27.448	24.669	28.356	27.668	25.313	27.595	24.289
	Time_G	35.833	20.207	29.865	29.642	20.456	34.099	25.105	33.962
	APSNR	21.170	23.763	21.596	24.259	24.600	21.774	23.079	21.131
LNOP [39]	PSNR_I	25.536	25.704	24.065	28.696	28.668	23.665	27.141	23.204
	Time_I	7.306	7.248	7.286	7.689	7.205	8.228	7.361	7.494
	PSNR_G	24.244	24.486	23.052	26.966	27.038	22.741	25.796	22.807
	Time_G	7.200	7.212	7.193	7.562	7.184	7.356	7.359	7.496
	APSNR	24.890	25.095	23.559	27.831	27.853	23.203	26.469	23.006
TTSVD [52]	PSNR_I	22.365	22.456	21.284	23.368	23.058	21.558	23.649	21.959
	Time_I	24.143	22.982	22.322	24.877	23.132	22.126	27.408	23.844
	PSNR_G	20.559	20.538	20.249	20.764	20.723	20.359	21.105	20.589
	Time_G	22.752	21.709	20.739	22.518	21.808	21.616	25.935	22.811
	APSNR	21.462	21.497	20.767	22.066	21.891	20.959	22.377	21.274
TNTV [51]	PSNR_I	21.201	23.068	21.511	26.202	25.349	17.826	26.106	21.321
	Time_I	16.259	16.544	16.195	16.474	16.325	16.098	16.230	16.093
	PSNR_G	21.095	22.076	21.774	24.086	24.408	19.772	24.253	22.241
	Time_G	16.948	16.065	17.362	16.586	16.305	16.789	17.229	16.205
	APSNR	21.148	22.572	21.643	25.144	24.879	18.799	25.180	21.781
ℓ_p-TTC [37]	PSNR_I	23.544	24.151	23.205	27.281	27.286	22.956	25.956	21.864
	Time_I	135.984	140.865	128.053	136.187	135.759	140.099	140.239	137.447
	PSNR_G	22.751	23.548	21.485	25.089	24.899	21.818	24.559	21.701
	Time_G	129.539	139.156	139.142	136.279	138.594	129.920	138.061	142.455
	APSNR	23.148	23.850	22.345	26.185	26.093	22.387	25.258	21.783
RTC-RE [54]	PSNR_I	25.127	25.494	23.840	27.748	27.792	23.762	26.544	23.518
	Time_I	41.658	41.697	51.660	40.696	39.378	46.342	41.923	44.159
	PSNR_G	23.590	23.777	22.784	25.096	25.087	22.712	24.668	22.715
	Time_G	51.848	49.057	51.357	51.025	51.003	51.230	50.952	45.992
	APSNR	24.358	24.636	23.312	26.422	26.440	23.237	25.606	23.117

elements. In the Gaussian noise scenarios, all the observed entries are not corrupted by anomalies, and thus a bigger ζ results in better recovery performance. Under impulsive noise, a small ζ leads to many entries to be mistaken as outliers, while a very large ζ cannot identify all the anomaly-contaminated entries. To achieve excellent performance in both types of noise, we select $\zeta = 3$ for the following experiments.

For the four observed images in Fig. 3, the restored pictures by CFN-RTC and its competitors are depicted in Fig. 5. The measurement metrics are listed below the corresponding recovered pictures. It is seen that the CFN-RTC attains the best performance on both the random and fixed masks in the presence of impulsive noise. In Gaussian noise, the performance of the CFN-RTC ranks second. It is worth pointing out that its performance is close to that of TRC adopting the Frobenius norm.

The effect of the percentage of missing pixels on performance is shown in Fig. 6 in which two types of noise are considered. The metric of APSNR signifies the average PSNR with Gaussian noise and impulsive noise. We see that the CFN-RTC attains better performance than TNTV, TRC, LNOP, TTSVD, ℓ_p-TTC, and RTC-RE at all the observation ratios. Note that the recovery accuracy of the RTC-RE severely decreases when the missing percentage increases.

Moreover, we compare all the algorithms with different outlier ratios. The experimental results for random mask with 50% observed data are depicted in Fig. 7. It is seen that the CFN-RTC outperforms the existing algorithms no matter the outlier ratio is large or small.

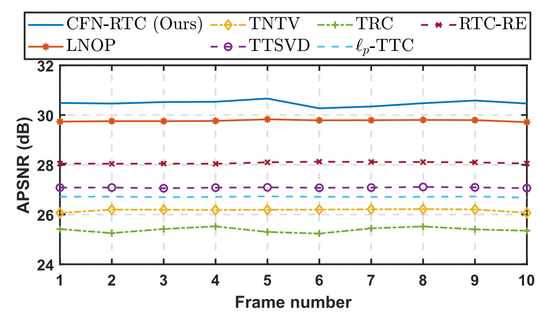

Fig. 10. APSNR of each frame on Akiyo with 50% randomly missing pixels.

Furthermore, eight well-known images, as shown in Fig. 8, are used to assess the inpainting performance. The results with 50% observation ratio for random mask are tabulated in Table II, while those of the deterministic mask are listed in Table III. It is seen that the CFN-RTC attains the highest PSNRs in the presence of impulsive noise, and better performance than TNTV, LNOP, TTSVD, ℓ_p-TTC, and RTC-RE in the Gaussian noise scenarios. Therefore, the average performance of the CFN-RTC with two types of noise is superior to all the competitors. It can be known that the capped Frobenius norm is able to attain comparable performance to the Frobenius norm in normal situation. On the other hand, the runtimes of the proposed method are less than those of ℓ_p-TTC and TRC-RE. Our approach and TRC are slower than LNOP, TTSVD, and TNTV since both of them adopt tensor ring decomposition. Although tensor ring factorization has a higher complexity than the t-SVD used by LNOP, TTSVD,

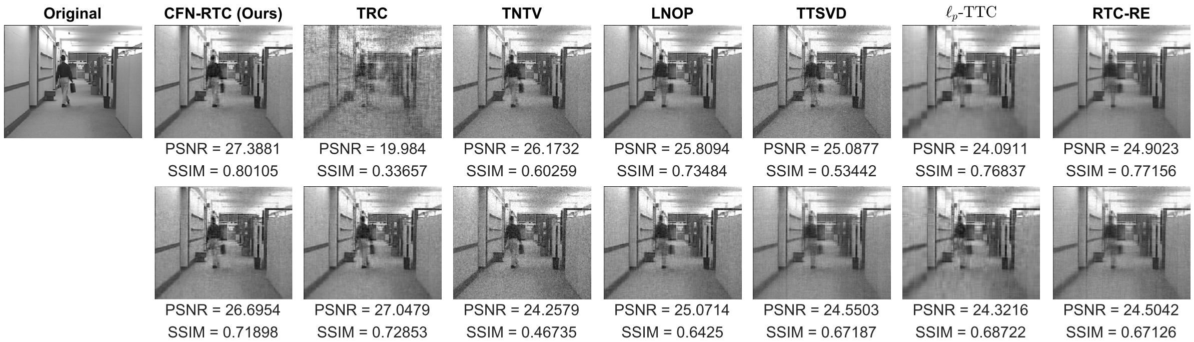

Fig. 11. Original and recovered frames for Hall. Top row corresponds to impulsive noise, while bottom row corresponds to Gaussian noise.

and TNTV, the former is able to handle higher order tensors, while the latter can only tackle the third-order tensors.

C. Video Restoration

The second application of TC is video restoration. Since LNOP, TTSVD, and TNTV only process the third-order tensors, we adopt grayscale videos to compare the CFN-RTC with the existing approaches. It is worth mentioning that the CFN-RTC is able to cope with higher order tensors. The examined dataset is YUV Video Sequences,[1] and we select two typical ones, namely, Akiyo and Hall. The dimensions of each frame are 147×176. We use the first ten frames of both the videos to assess different algorithms, which is adopted in [39]. Thereby, the dimensions of each video are $147 \times 176 \times 10$. The recovery performance is evaluated using PSNR and SSIM.

Fig. 9 shows one of recovered frames of the Akiyo video under 50% randomly missing pixels. The first row shows the results with impulsive noise, while the second row depicts the restored frames with Gaussian noise where Gaussian and impulsive components are the same as the previous settings. We see that the CFN-RTC achieves higher PSNR and SSIM values than TRC, TNTV, LNOP, TTSVD, ℓ_p-TTC, and RTC-RE in the presence of impulsive noise. In Gaussian noise, the CFN-RTC and TRC attain better performance than their competitors. However, the average performance of the CFN-RTC is the best among seven algorithms. Fig. 10 shows the plots of the average performance of all the frames. It is seen that the APSNRs of the CFN-RTC are larger than those of the other approaches.

Under the same condition as the Akiyo video, the results of the Hall video are shown in Figs. 11 and 12. Fig. 11 shows one of restored frames, while Fig. 12 shows the plots of the average performance of all the frames. It is seen that the CFN-RTC outperforms its competitors in impulsive noise. Although its performance ranks second in Gaussian noise, its average performance is superior to the competing algorithms in Fig. 12.

Furthermore, we investigate the recovery performance of different algorithms under a high percentage of missing pixels, namely, 80% randomly missing pixels. The experimental results are plotted in Figs. 13 and 14. It is seen that the suggested method still attains better performance than its competitors under a high missing percentage. Note that the performance of the RTC-RE severely degrades when the missing percentage increases.

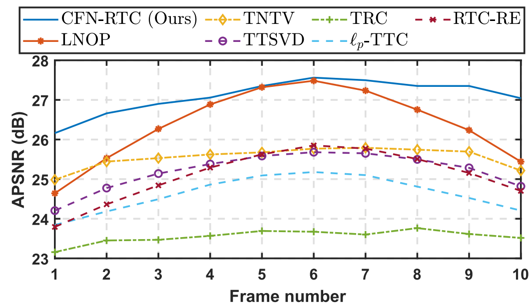

Fig. 12. APSNR of each frame on Hall with 50% randomly missing pixels.

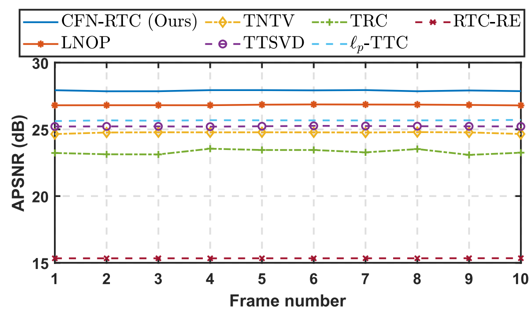

Fig. 13. APSNR of each frame on Akiyo with 80% randomly missing pixels.

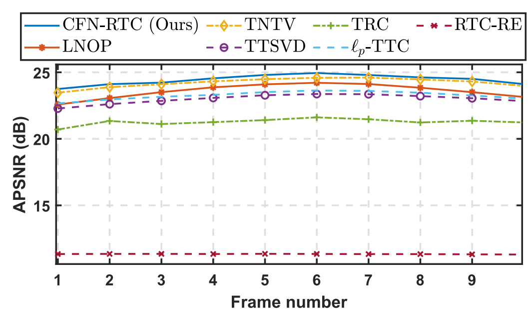

Fig. 14. APSNR of each frame on Hall with 80% randomly missing pixels.

[1] http://trace.eas.asu.edu/yuv/

V. Conclusion

In this article, we have devised an RTC algorithm using the capped Frobenius norm and tensor ring decomposition. The upper bound of the capped Frobenius norm is automatically updated using the normalized median absolute deviation strategy. The half-quadratic theory is used to simplify the nonconvex problem, resulting in a tractable task such that it becomes a convex optimization w.r.t. each individual variable. Then the PBCD method is exploited to handle the resultant problem, yielding an algorithm called CFN-RTC. The convergence behavior of the CFN-RTC is analyzed, that is, the objective function value is guaranteed to be convergent while the variable sequence has a subsequence to converge to a critical point. The experimental results on real-world images and videos demonstrate that the CFN-RTC achieves higher recovery accuracy than six popular algorithms in the presence of impulsive noise. Besides, its performance is comparable to the Frobenius-norm-based method without tweaking parameter in Gaussian noise.

References

[1] P. Comon, "Tensors: A brief introduction," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 44–53, May 2014.

[2] L. Yang, Z.-H. Huang, and X. Shi, "A fixed point iterative method for low n-rank tensor pursuit," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2952–2962, Jun. 2013.

[3] Y. Liu and F. Shang, "An efficient matrix factorization method for tensor completion," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 307–310, Apr. 2013.

[4] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[5] Q. Liu, F. Davoine, J. Yang, Y. Cui, Z. Jin, and F. Han, "A fast and accurate matrix completion method based on QR decomposition and $L_{2,1}$-norm minimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 803–817, Mar. 2018.

[6] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Logarithmic norm regularized low-rank factorization for matrix and tensor completion," *IEEE Trans. Image Process.*, vol. 30, pp. 3434–3449, 2021.

[7] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, May 2017.

[8] T.-H. Chou, N. Michelusi, D. J. Love, and J. V. Krogmeier, "Fast position-aided MIMO beam training via noisy tensor completion," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 774–788, Apr. 2021.

[9] A. J. Tom and S. N. George, "Video completion and simultaneous moving object detection for extreme surveillance environments," *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 577–581, Apr. 2019.

[10] I. Kajo, N. Kamel, and Y. Ruichek, "Incremental tensor-based completion method for detection of stationary foreground objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 5, pp. 1325–1338, May 2019.

[11] I. Kajo, N. Kamel, and Y. Ruichek, "Tensor-based approach for background-foreground separation in maritime sequences," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7115–7128, Nov. 2021.

[12] Y. Chen, T. Huang, W. He, N. Yokoya, and X. Zhao, "Hyperspectral image compressive sensing reconstruction using subspace-based nonlocal tensor ring decomposition," *IEEE Trans. Image Process.*, vol. 29, pp. 6813–6828, 2020.

[13] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, nos. 1–4, pp. 164–189, Apr. 1927.

[14] R. Bro, "PARAFAC. Tutorial and applications," *Chemometrics Intell. Lab. Syst.*, vol. 38, no. 2, pp. 149–171, Oct. 1997.

[15] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sep. 2015.

[16] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S.-I. Amari, "Bayesian robust tensor factorization for incomplete multiway data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 736–748, Apr. 2015.

[17] Y. Liu, Z. Long, H. Huang, and C. Zhu, "Low CP rank and tucker rank tensor completion for estimating missing components in image data," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 4, pp. 944–954, Apr. 2020.

[18] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Feb. 1966.

[19] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Hyperspectral computational imaging via collaborative Tucker3 tensor decomposition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 98–111, Jan. 2021.

[20] J. Xue, Y. Zhao, S. Huang, W. Liao, J. C.-W. Chan, and S. G. Kong, "Multilayer sparsity-based tensor decomposition for low-rank tensor completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6916–6930, Nov. 2022.

[21] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[22] W. Sun, L. Huang, H. C. So, and J. Wang, "Orthogonal tubal rank-1 tensor pursuit for tensor completion," *Signal Process.*, vol. 157, pp. 213–224, Apr. 2019.

[23] K. Gilman, D. A. Tarzanagh, and L. Balzano, "Grassmannian optimization for online tensor completion and tracking with the t-SVD," *IEEE Trans. Signal Process.*, vol. 70, pp. 2152–2167, 2022.

[24] Q. Jiang and M. Ng, "Robust low-tubal-rank tensor completion via convex optimization," in *Proc. 28th Int. Joint Conf. Artif. Intell., (IJCAI)*, Macao, China, Aug. 2019, pp. 2649–2655.

[25] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, Sep. 2011.

[26] L. Yuan, Q. Zhao, L. Gui, and J. Cao, "High-order tensor completion via gradient-based optimization under tensor train format," *Signal Process., Image Commun.*, vol. 73, pp. 53–61, Apr. 2019.

[27] Y. Zhang, Y. Wang, Z. Han, X. Chen, and Y. Tang, "Effective tensor completion via element-wise weighted low-rank tensor train with overlapping ket augmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7286–7300, Nov. 2022.

[28] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," 2016, *arXiv:1606.05535*.

[29] W. Wang, V. Aggarwal, and S. Aeron, "Efficient low rank tensor ring completion," in *Proc. ICCV*, Venice, Italy, Oct. 2017, pp. 5697–5705.

[30] W. He, N. Yokoya, L. Yuan, and Q. Zhao, "Remote sensing image reconstruction using tensor ring completion and total variation," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8998–9009, Nov. 2019.

[31] H. Huang, Y. Liu, J. Liu, and C. Zhu, "Provable tensor ring completion," *Signal Process.*, vol. 171, Jun. 2020, Art. no. 107486.

[32] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, Q. Zhao, and T.-X. Jiang, "Fully-connected tensor network decomposition and its application to higher-order tensor completion," in *Proc. Conf. AAAI Artif. Intell.*, May 2021, vol. 35, no. 12, pp. 11071–11078.

[33] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge, U.K.: Cambridge Univ. Press, 2018.

[34] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1479–1485, Oct. 2005.

[35] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts," *IEEE Signal Process. Mag.*, vol. 29, no. 4, pp. 61–80, Jul. 2012.

[36] W. Sun, X. Lin, H. C. So, L. Huang, and Q. Li, "Iteratively reweighted tensor SVD for robust multi-dimensional harmonic retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 4318–4322.

[37] Q. Liu, X. Li, H. Cao, and Y. Wu, "From simulated to visual data: A robust low-rank tensor completion approach using ℓ_p-regression for outlier resistance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3462–3474, Jun. 2022.

[38] S. A. Vorobyov, Y. Rong, N. D. Sidiropoulos, and A. B. Gershman, "Robust iterative fitting of multilinear models," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2678–2689, Aug. 2005.

[39] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Robust low-rank tensor recovery via nonconvex singular value minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 9044–9059, 2020.

[40] X. P. Li and H. C. So, "Robust low-rank tensor completion based on tensor ring rank via $\ell_{p,\epsilon}$-norm," *IEEE Trans. Signal Process.*, vol. 69, pp. 3685–3698, 2021.

[41] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, Jul. 2018.

[42] B.-Z. Li, X.-L. Zhao, J.-L. Wang, Y. Chen, T.-X. Jiang, and J. Liu, "Tensor completion via collaborative sparse and low-rank transforms," *IEEE Trans. Comput. Imag.*, vol. 7, pp. 1289–1303, 2021.

[43] H. Huang, Y. Liu, Z. Long, and C. Zhu, "Robust low-rank tensor ring completion," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1117–1126, 2020.

[44] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA, USA: MIT Press, 1987.

[45] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1127–1134, Apr. 2020.

[46] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Signal Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.

[47] H. T. Shen, Y. Zhu, W. Zheng, and X. Zhu, "Half-quadratic minimization for unsupervised feature selection on incomplete data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 7, pp. 3122–3135, Jul. 2021.

[48] F. Wen, R. Ying, P. Liu, and T.-K. Truong, "Nonconvex regularized robust PCA using the proximal block coordinate descent algorithm," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5402–5416, Oct. 2019.

[49] L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao, "Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion," in *Proc. Conf. AAAI Artif. Intell.*, Jan. 2019, vol. 33, no. 1, pp. 9151–9158.

[50] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. IEEE CVPR*, Columbus, OH, USA, Jun. 2014, pp. 3842–3849.

[51] D. Qiu, M. Bai, M. K. Ng, and X. Zhang, "Robust low-rank tensor completion via transformed tensor nuclear norm with total variation regularization," *Neurocomputing*, vol. 435, pp. 197–215, May 2021.

[52] G. Song, M. K. Ng, and X. Zhang, "Robust tensor completion using transformed tensor singular value decomposition," *Numer. Linear Algebra Appl.*, vol. 27, no. 3, p. e2299, Mar. 2020.

[53] Q. Yao, J. T. Kwok, T. Wang, and T.-Y. Liu, "Large-scale low-rank matrix learning with nonconvex regularizers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2628–2643, Nov. 2019.

[54] Q. Shi, Y.-M. Cheung, and J. Lou, "Robust tensor SVD and recovery with rank estimation," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10667–10682, Oct. 2022.

[55] C. Guillemot and O. Le Meur, "Image inpainting : Overview and recent advances," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 127–144, Jan. 2014.

[56] X. P. Li, M. Wang, and H. C. So, "An interpretable bi-branch neural network for matrix completion," *Signal Process.*, vol. 200, Nov. 2022, Art. no. 108640.

[57] X. P. Li, Z.-L. Shi, Q. Liu, and H. C. So, "Fast robust matrix completion via entry-wise ℓ_0-norm minimization," *IEEE Trans. Cybern.*, early access, Dec. 6, 2022, doi: 10.1109/TCYB.2022.3224070.

Xiao Peng Li received the B.Eng. degree in electronic science and technology from Yanshan University, Qinhuangdao, China, in 2015, and the M.Sc. degree (Hons.) in electronic information engineering and the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, SAR, China, in 2018 and 2022, respectively.

He is currently a Post-Doctoral Fellow with the Department of Electrical Engineering, City University of Hong Kong. His research interests include optimization methods, machine learning, sparse recovery, matrix processing, and tensor processing.

Zhi-Yong Wang was born in Henan, China. He received the B.S. degree in mechanical engineering from Zhengzhou University, Zhengzhou, China, in 2017, and the M.S. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, SAR, China, supervised by Prof. Hing Cheung So.

His research interests include sparse recovery, robust signal processing, and low-rank approximation.

Zhang-Lei Shi received the Ph.D. degree from the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, SAR, China, in 2021.

He is currently a Lecturer with the College of Science, China University of Petroleum (East China), Qingdao, China. His current research interests include neural networks and machine learning.

Hing Cheung So (Fellow, IEEE) was born in Hong Kong. He received the B.Eng. degree in electronic engineering from the City University of Hong Kong, Hong Kong, SAR, China, 1990, and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 1995.

From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. From 1995 to 1996, he was a Post-Doctoral Fellow with The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor. His research interests include detection and estimation, multidimensional harmonic retrieval, robust signal processing, and sparse approximation.

Dr. So has been on the editorial boards of *IEEE Signal Processing Magazine* (2014–2017), IEEE TRANSACTIONS ON SIGNAL PROCESSING (2010–2014), *Signal Processing* (2010), and *Digital Signal Processing* (2011). He was also a Lead Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, Special Issue on "Advances in Time/Frequency Modulated Array Signal Processing" in 2017.

Nicholas D. Sidiropoulos (Fellow, IEEE) received the Diploma degree in electrical engineering from the Aristotle University of Thessaloniki, Thessaloniki, Greece, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 1990 and 1992, respectively.

He is currently a Louis T. Rader Professor at the University of Virginia, Charlottesville, VA, USA. He has previously served on the faculty of the University of Minnesota, Minneapolis, MN, USA, and the Technical University of Crete, Kounoupidiana, Greece. His research interests are in signal processing, communications, optimization, and tensor decomposition with applications in machine learning and communications.

Dr. Sidiropoulos is a fellow of EURASIP in 2014. He received the NSF/CAREER Award in 1998, the IEEE Signal Processing Society (SPS) Best Paper Award in 2001, 2007, 2011, and 2023, and the IEEE SPS Donald G. Fink Overview Paper Award in 2023. He received the IEEE Signal Processing Society Meritorious Service Award in 2010, the Distinguished Alumni Award of the Department of ECE, University of Maryland in 2013, the EURASIP Technical Achievement Award in 2022, and the IEEE SPS Claude Shannon–Harry Nyquist Technical Achievement Award in 2023. He served as the IEEE SPS Distinguished Lecturer (2008–2009), as the Vice President-Membership of IEEE SPS (2017–2019), and as the Chair for the SPS IEEE Fellow Evaluation Committee (2020–2021).